DL team15 Final Project Report

**0.Title Name**

CAPTCHA Image Recognition

**1. Members list**

|  |  |  |
| --- | --- | --- |
| Student ID | Name | Work Assignment |
| 0513403 | 陳昀萱 | Preprocessing, write report |
| 0513471 | 陳姿妤 | Model Implementation, write report |
| 0513419 | 范舒淇 | Preprocessing, write report |
| 0853402 | 吳奐萱 | Model Implementation, write report |
| 0853420 | 林若瑜 | Model Implementation, write report |

**2. Introduction/Motivation**

**Introduction.**

We would like to identify the characters of Captcha on Kaggle’s dataset and New E3.

We would use the dataset on the Kaggle to get a great sense of how to solve this problem first, and then moving on to solve the Captcha problem on New E3. By solving the Captcha problem on New e3, we can deliver a great time-saving solution for ourselves and our fellow classmates.

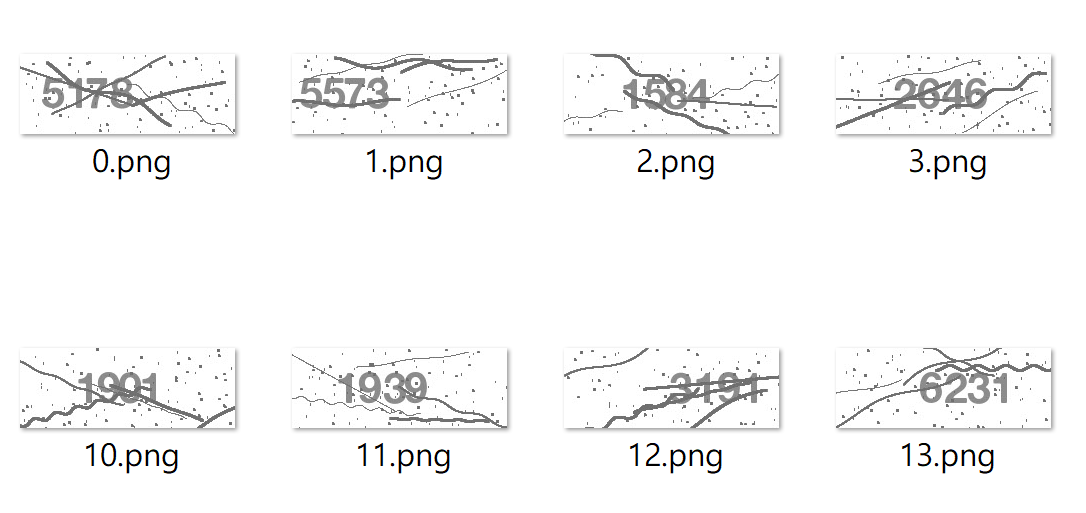
**Motivation.**

Entering the captcha on New e3 every day is a pretty annoying problem! If we spend 10 minutes entering captcha on New e3 every day, we would spend 70 minutes per week, 280 minutes per month on this tedious task! Therefore, if we can solve this problem by using state-of-art deep learning models we learn in the class, it would be both a meaningful and practical project to research on.

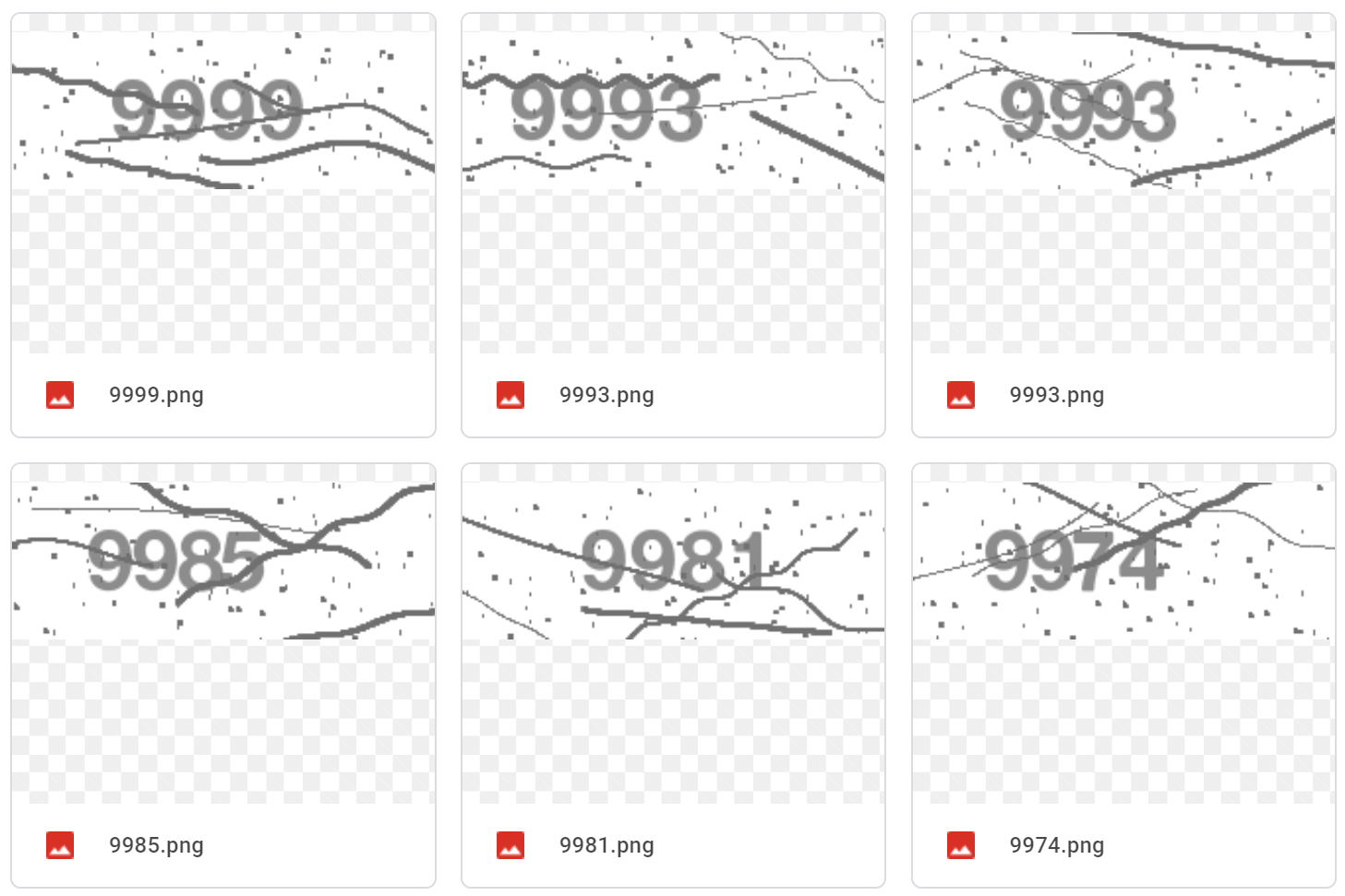
**3. Technical Part: Data\_Preprocess/CNN/DNN/VE**

* 取得 dataset: Kaggle & Crawling from New E3

1. New E3

First, we used the python module, selenium, to open a New E3 website by webdriver, and crawled the captcha in the login site. The captcha id is 'captcha-desktop', and we downloaded this element with 'urllib.request' module. After that, we got 3000 pieces of captchas as our New E3 dataset.

Second, we renamed the downloaded captchas with the numbers in the picture. Then the name of pictures can be used as our labels in the process of constructing the model.



After viewing screenshots taken from the New E3 system, we noticed that although all the dots, lines, and numbers are gray, they are in different scales of gray. Using a color detecting application, we discovered the number part of the image was (140, 140, 140) in RGB scale. Therefore, our first step was to extract that particular color out. After denoising, we modified it to a black and white image with only the number part.

|  |  |
| --- | --- |
|  |  |
| 1. Screenshot image from New E3 | 1. Extract RGB(140, 140, 140) |
|  |  |
| 1. After Denoising | 1. Change to black and white |

2. Kaggle dataset

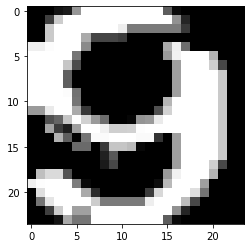
For Kaggle dataset, we only turned the raw image into black and white.

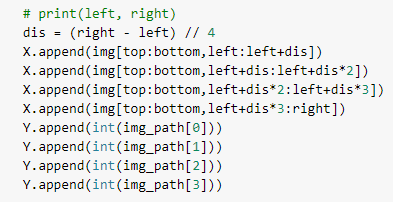
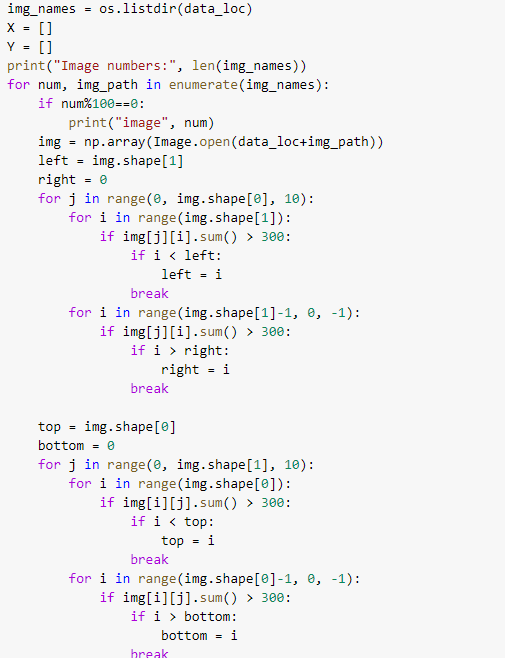
|  |  |
| --- | --- |
|  |  |
| Raw Image | Black and White Image |

* CNN/ DNN Model 資料前處理：
  + 分成兩部分(使用原始data和把原始data切成四個單一數字)
  + 1.原本直接使用整張圖（ 215\*80 pixel ) 去 train，但準確率極低，因此改將資料切分成一個一個的數字訓練，效果好非常多（詳細結果在第四部分）。

 ← 整張圖（215\*80 pixel)示意圖

* + 2.將原是圖片切分出單一數字（ 24\*24 pixel ）部分訓練。

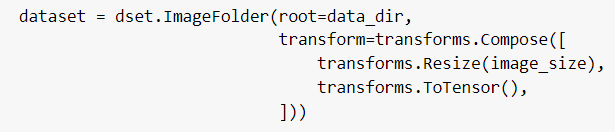
 ← 整張圖（ 24\*24 pixel )示意圖

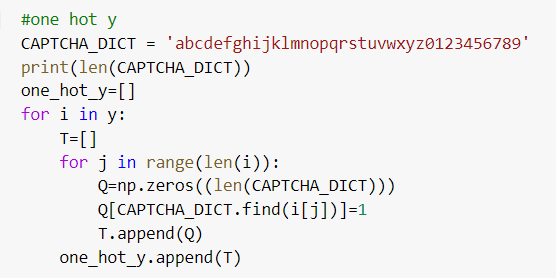
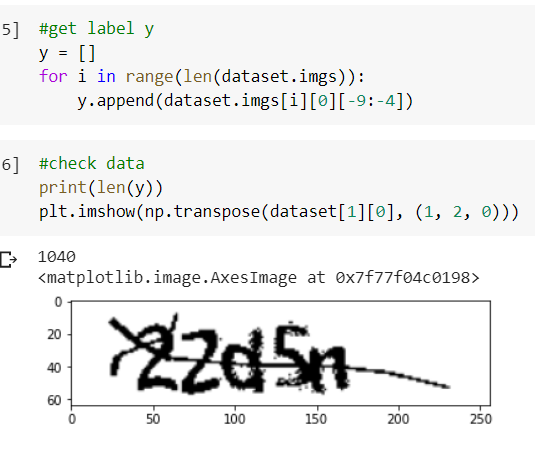


將圖片中數字的位置放入 X, Y list中，方便未來取出數字。

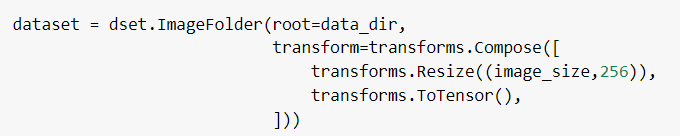
* AE Model 資料前處理：
  + 這次實驗測試了kaggle和e3的資料，都將圖片 reshape 成 64\*256，之後再下去training。
  + 透過讀取檔案名稱當成target，之後再one-hot encoding。

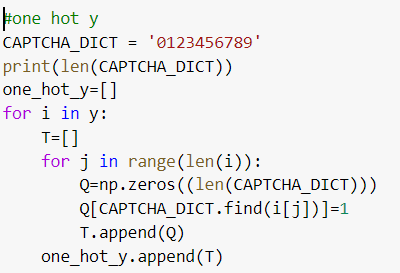
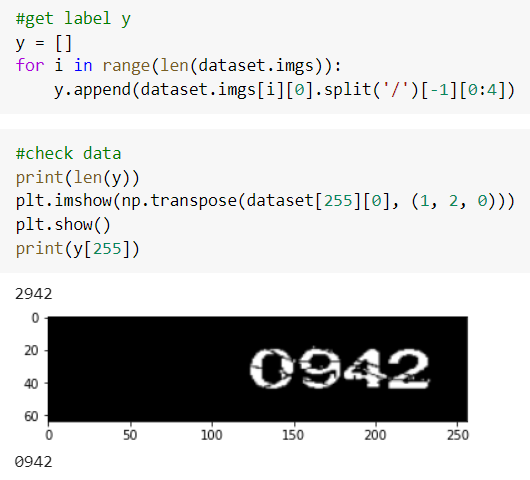
kaggle



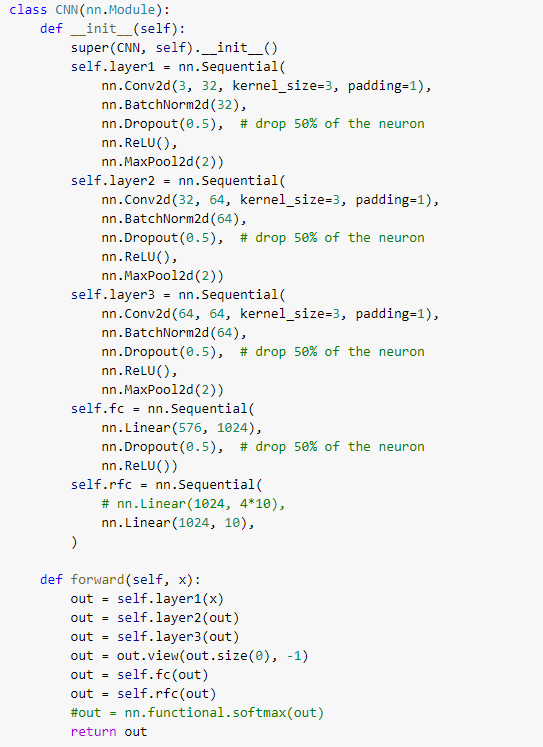


e3





* CNN Model 建立：

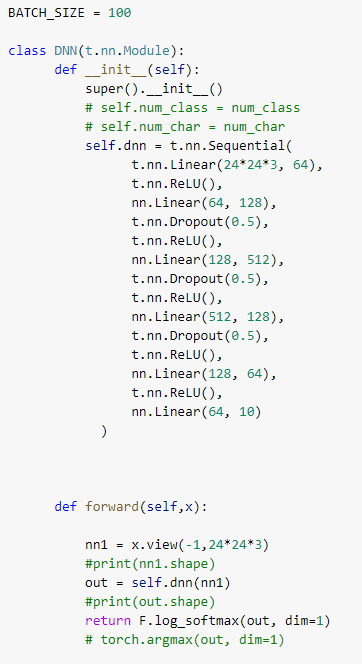


input = 24\*24\*3

output = 10(0~9)

activation function = ReLU

* DNN Model 建立：



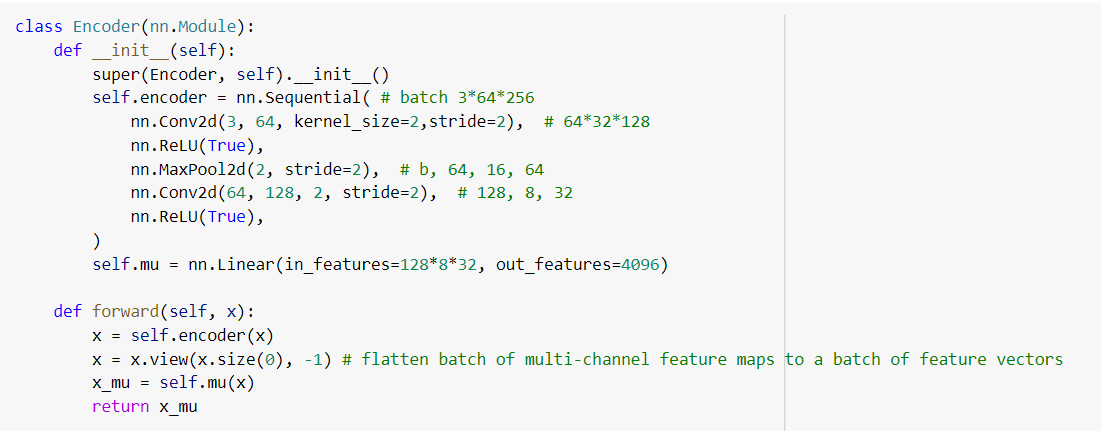
input = 24\*24\*3

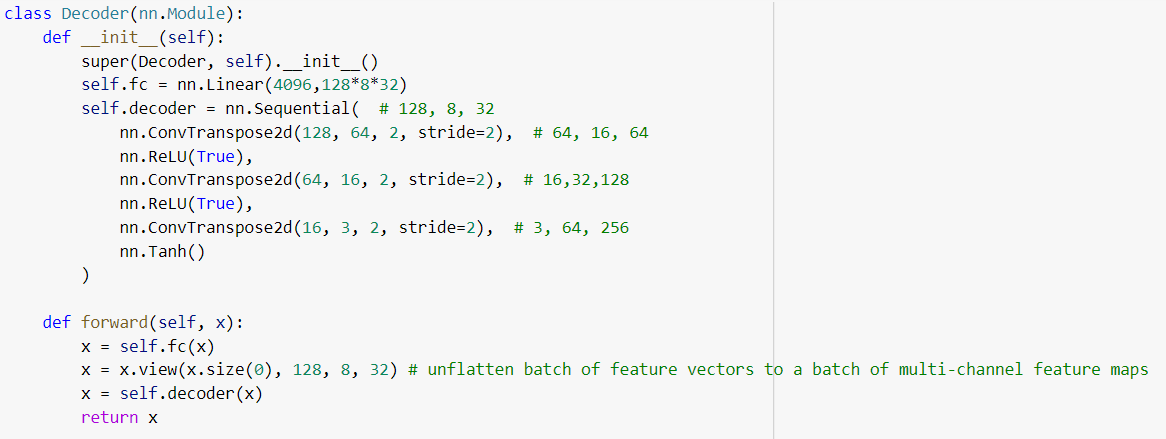
output = 10(0~9)

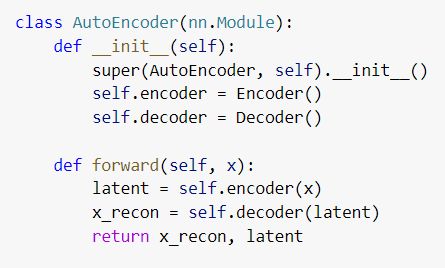
activation function = ReLU

* AE Model 建立：

kaggle



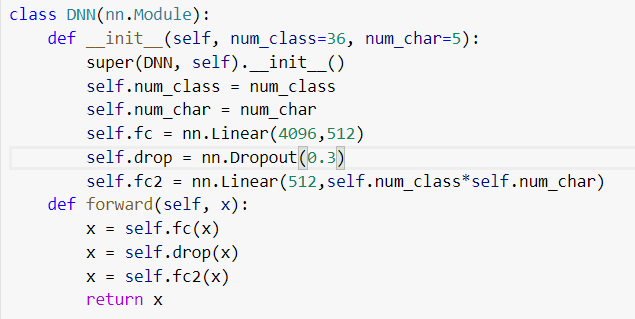




input: 3\*64\*256 (batch size= 64)

output: 4096 (latent vector)

activation: Tanh()



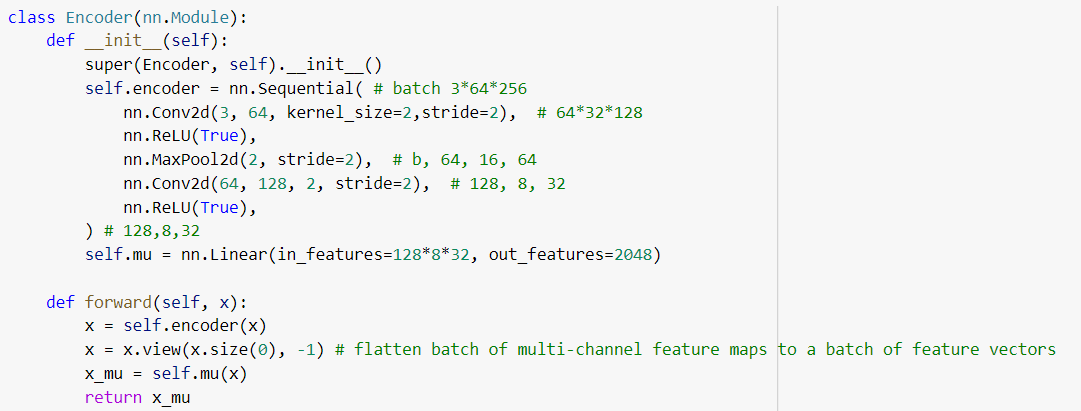
for predict result:

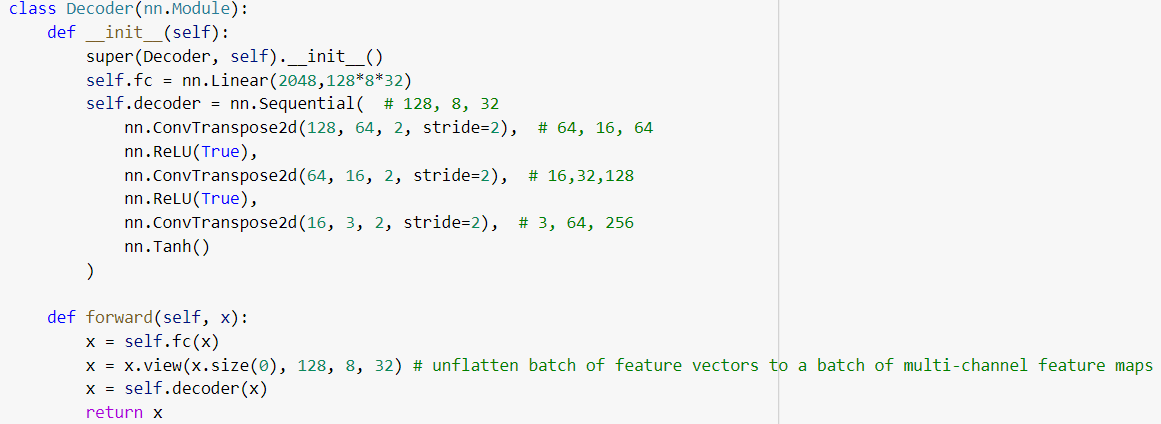
input: 4096 (latent vector)

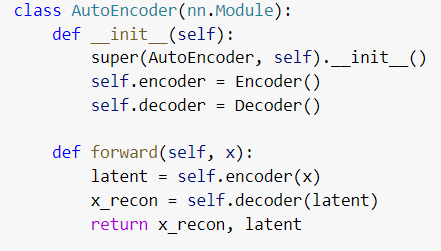
output: 36\*5 (one-hot result)

activation: None

e3



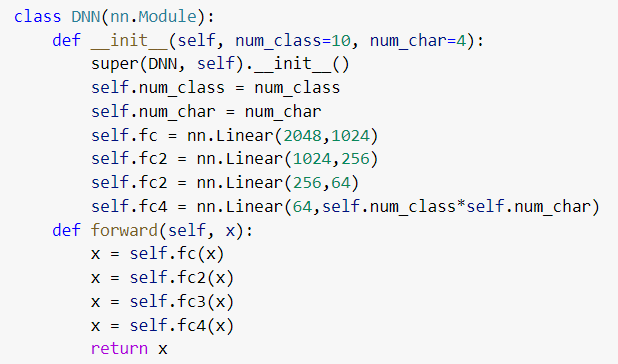




input: 3\*64\*256 (batch size=64)

output = 2048 (latent vector)

activation = Tanh()



for predict result:

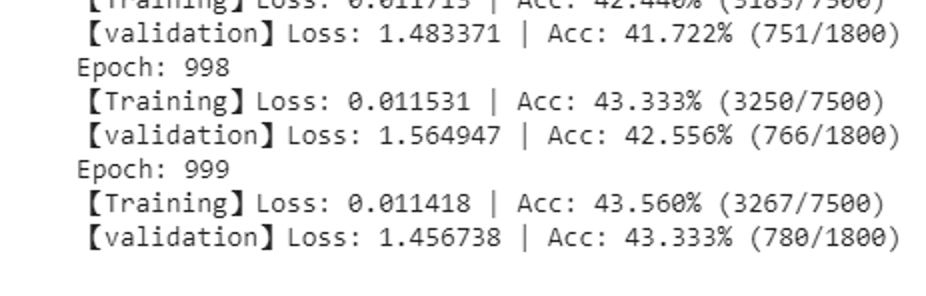
input: 2048 (latent vector)

output: 10\*4 (one-hot result)

activation: None

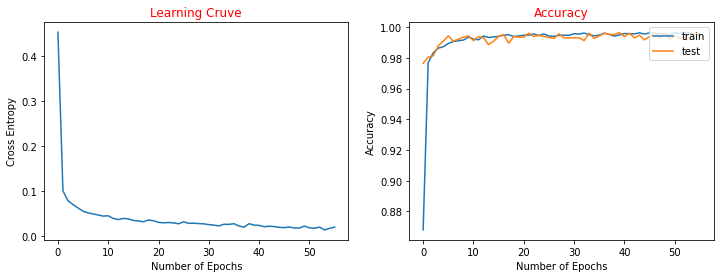
**4. Experiment**

* 一開始使用原始dataset(整張圖-四個數字)的結果，不太理想：



因此我們決定把資料再切每一張照片裡一個數字去做判斷，就是變成一般的單一數字辨識。

* CNN Model 結果：
  + batch\_size = 100, 跑 50 個 epoch
  + 此處的 test 為單一數字辨識的準確率，可以達到 98%



* + 以下為四個數字的準確率(四個都一樣才算對)

target: tensor([8., 3., 5., 2.]) predict: tensor([8., 3., 5., 2.])

target: tensor([8., 3., 4., 7.]) predict: tensor([8., 3., 4., 7.])

target: tensor([8., 7., 2., 6.]) predict: tensor([8., 7., 2., 6.])

target: tensor([1., 7., 1., 3.]) predict: tensor([1., 7., 1., 3.])

target: tensor([4., 2., 1., 8.]) predict: tensor([4., 2., 1., 8.])

target: tensor([3., 7., 6., 7.]) predict: tensor([3., 7., 6., 7.])

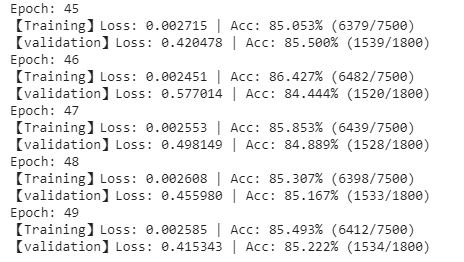
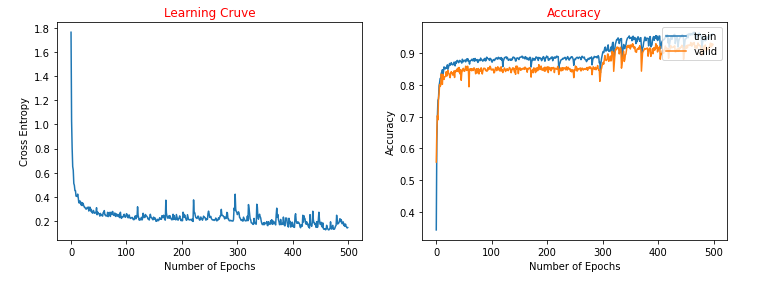
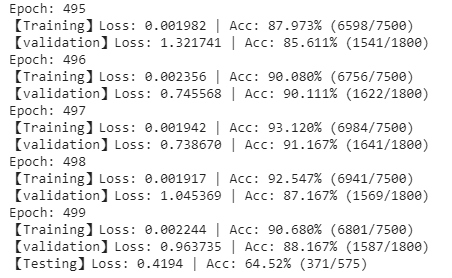
target: tensor([0., 9., 7., 0.]) predict: tensor([0., 9., 7., 0.])

target: tensor([4., 4., 0., 0.]) predict: tensor([4., 4., 0., 0.])

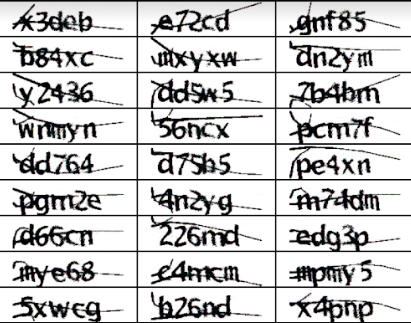
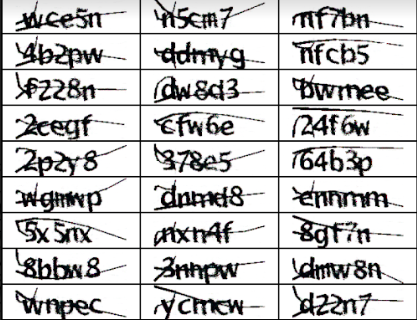
target: tensor([8., 1., 4., 7.]) predict: tensor([8., 1., 4., 7.])

target: tensor([7., 0., 0., 5.]) predict: tensor([7., 0., 0., 5.])

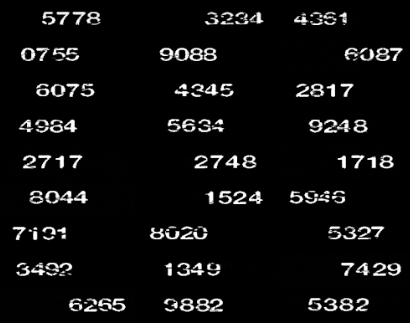
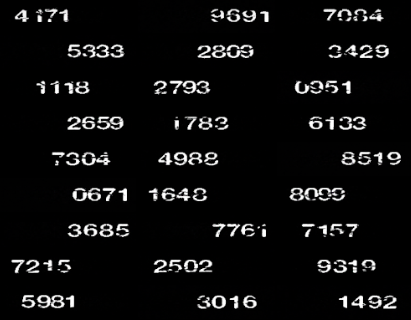
target: tensor([6., 2., 6., 5.]) predict: tensor([6., 2., 6., 5.])

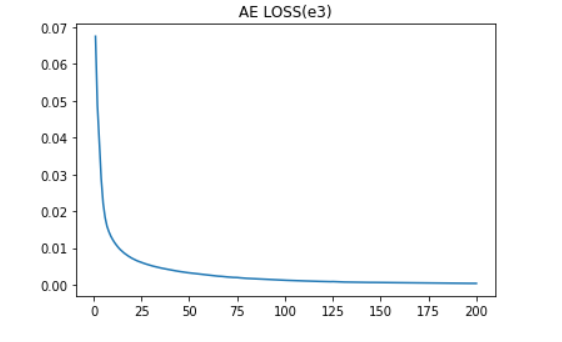
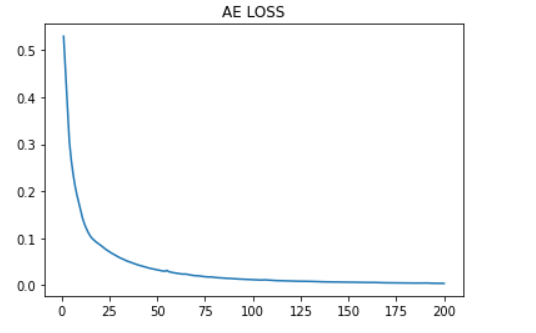
* DNN Model 結果：
  + batch\_size = 100, 跑 50 個 epoch
  + 相較於CNN的 98%，DNN表現較差，train準確度最高只到達 85%，因此打算多跑幾個 epoch。
  + batch\_size = 100, 跑 500 個 epoch
  + 相比跑 50 個 epoch 時 準確度 85%，跑 500 個 epoch 準確度 93%
  + 
* AE Model 結果：
  + 分成兩部分，第一個部分為透過AE取出latent vector，之後使用latent vector 下去做預測結果。
  + batch\_size = 64, epoch=200

kaggle

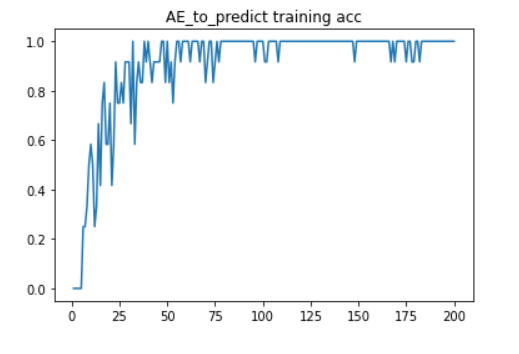
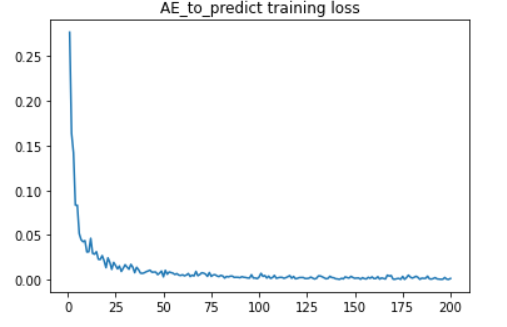


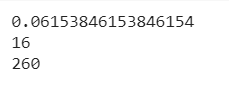
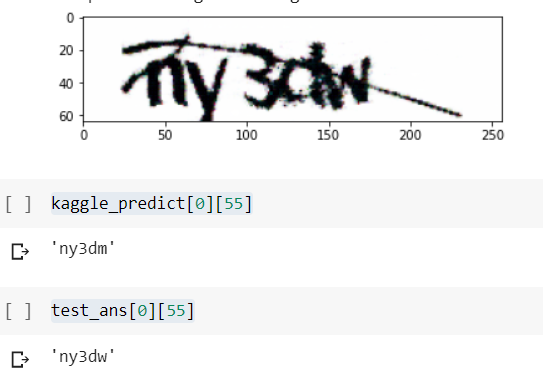
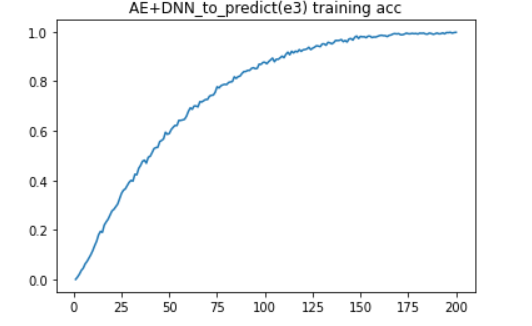
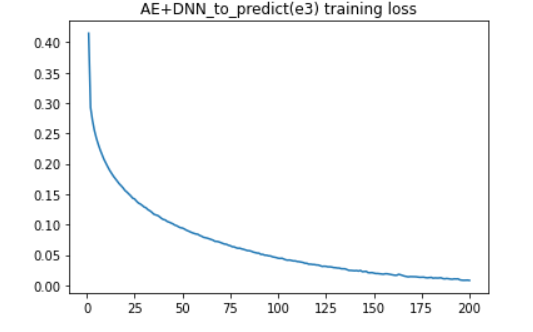
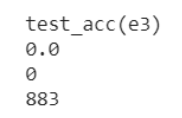
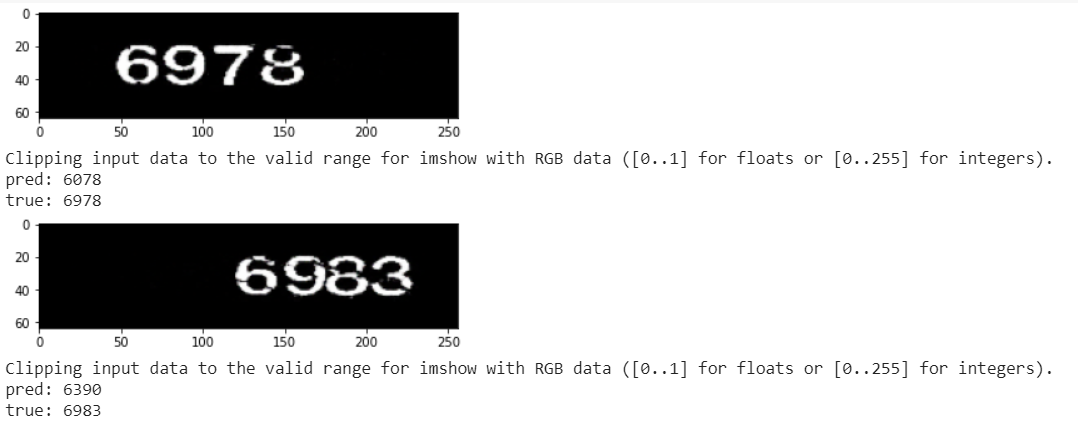
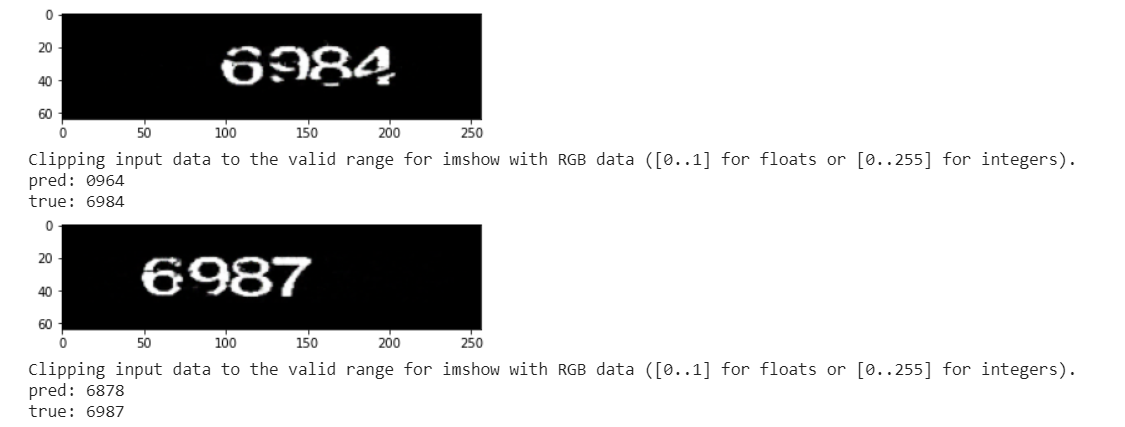
e3





* + 就AE MODEL來看，可以看到兩個模型Loss都很低且還原的程度跟原本的程度差不多，所以原本以為最後output出來的latent vector足夠代表圖片本身。
  + 所以接下來第二部分，將使用這些latent vector放入DNN MODEL去預測出最後的結果(數字)出來。
  + 首先介紹kaggle的預測結果



* + 雖然training loss 和 accuracy都很好，但最後實驗在test 的結果卻十分差勁，幾乎沒有正確性可言。
  + 下面是test 預測結果，accuracy只有0.06，對了16個，總共有260筆資料。
  + 
  + 
  + 上圖為少數正確的結果，有印了幾張圖片出來，但每次幾乎都只猜對4個總是會錯1個字母or數字，而無法完全正確。
  + 接下來是e3的介紹結果
  + 
  + 這邊的結果更加慘烈，實驗過latent vector = 4096,2048,1024,512版本，每一個出來的結果都十分差勁，而且比kaggle結果來的更糟糕，雖然同樣training 表現不錯。
  + 
  + 這邊的test 結果甚至一個都沒有答對
  + 
  + 
  + 且相較前者kaggle只有錯一個字母，這邊甚至長長錯的不只一個字母，結果十分慘淡，因此推測出或許AE MODEL不太適合用來預測出數字結果，或許學出來的圖片特徵具有代表性可以還原出原本圖片，但不是合用來預測結果，這邊猜測有一個原因應該是因為圖片本身為了還原回去，還有不是數字的地方是不需要的，因此產生了干擾。且因為４個數字擺在一起，或許學出來的特徵其實把這幾個數字纏在一起也說不定，因此也會造成干擾，像前面的兩位同學所實作的ＤＮＮ和ＣＮＮ就有將數字分離出來再來預測結果，結果就表現十分良好，因此認為會許分割出數字再來預測結果會是比較好的作法。

**5. Reference**

**Dataset.**

* <https://www.kaggle.com/fournierp/captcha-version-2-images>
* screenshots captured on the New e3 system.

**Paper.**

1. Lu, Y.: Machine printed character segmentation; an overview. Pattern Recognition 28(1), 67–80 (1995)
2. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: 2013 12th International Conference on Document Analysis and Recognition. vol. 2, pp. 958–958. IEEE Computer Society (2003)
3. Jaderberg M., Vedaldi A., Zisserman A. (2014) Deep Features for Text Spotting. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8692. Springer, Cham
4. CAPTCHA recognition based on deep convolutional neural network Jing Wang, Jiaohua Qin \*, Xuyu Xiang, Yun Tan and Nan Pan College of Computer Science and Information Technology, Central South University of Forestry and Technology, 498 shaoshan S Rd, Changsha, 410004, China
5. Korakakis, Michalis & Magkos, Emmanouil & Mylonas, Phivos. (2014). Automated CAPTCHA Solving: An Empirical Comparison of Selected Techniques. Proceedings - 9th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2014. 44-47. 10.1109/SMAP.2014.29.